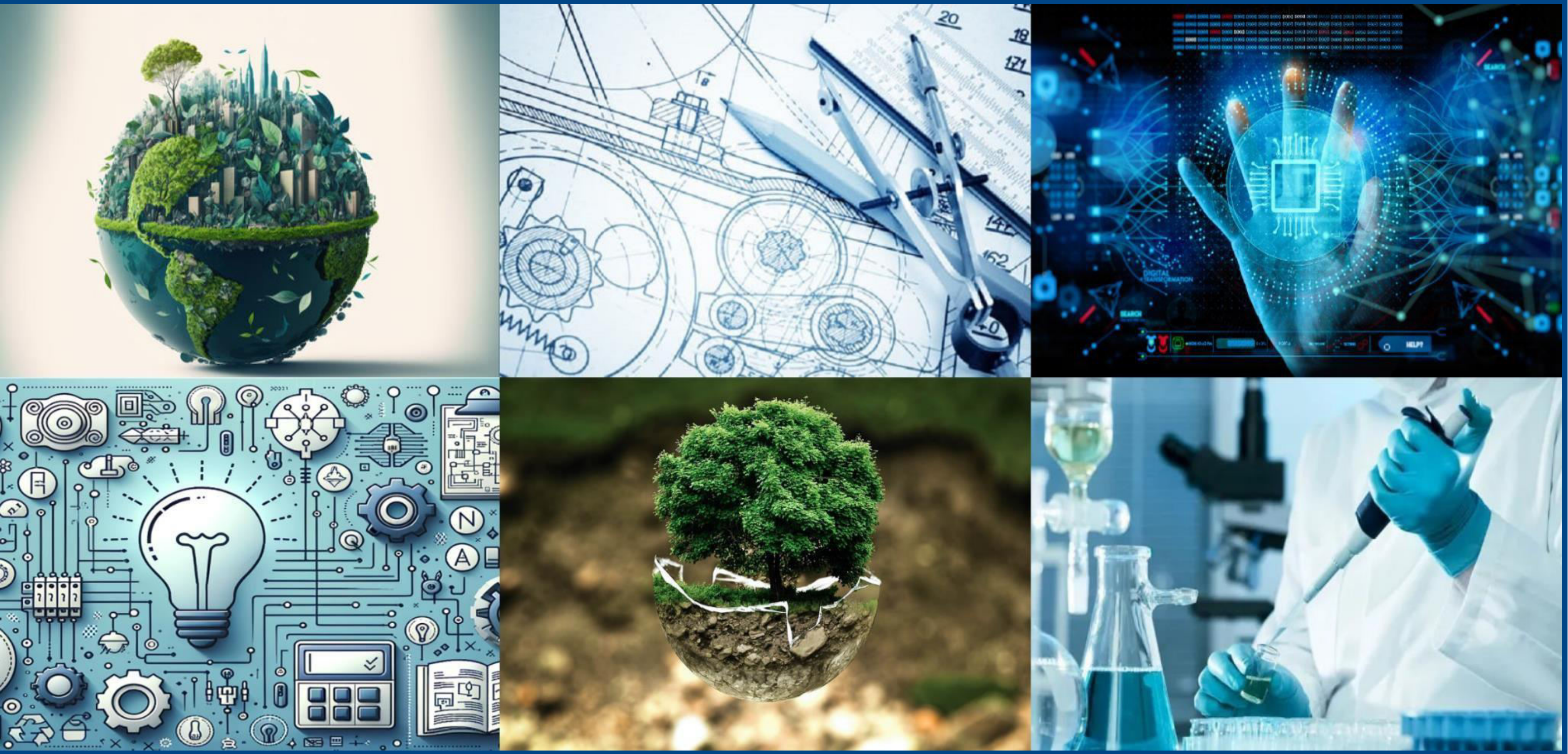
ISSN: 2582-7219

# International Journal of Multidisciplinary Research in Science, Engineering and Technology

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



Impact Factor: 8.206

Volume 8, Issue 7, July 2025

**© 2025 IJMRSET | Volume 8, Issue 7, July 2025| DOI:10.15680/IJMRSET.2025.0807048**

**ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|**

**International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)**

**(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)**

# Image Steganography using Python

**Ms. S. Abirami[1], Mrs. P. Shenbagam[2]**

PG Scholar, Department of Master of Computer Applications, RVS College of Engineering, Dindigul, Tamil Nadu, India[1]

Assistant Professor, Department of Master of Computer Applications, RVS College of Engineering, Dindigul, Tamil Nadu, India[2]

**ABSTRACT:** In the realm of secure communication and data protection, steganography plays a pivotal role by concealing information within seemingly innocuous carrier files such as images. This paper presents a Python application utilizing Tkinter for the graphical user interface (GUI) and the Python Imaging Library (PIL) for image processing, aimed at implementing image steganography. The proposed system allows users to embed secret messages into digital images while maintaining the visual integrity of the carrier image.

Additionally, it provides functionalities for extracting hidden messages from steganographic images. Through a user-friendly interface, users can select images and encode/decode messages seamlessly, enhancing the accessibility and usability of steganographic techniques. The implementation demonstrates the effectiveness of Python in developing practical solutions for data security and privacy, offering a versatile tool for individuals and organizations to safeguard sensitive information through covert communication channels.

## I. INTRODUCTION

In today's digital era, where data security and privacy have become critical concerns, steganography emerges as a powerful technique for secure communication. Unlike cryptography, which focuses on encrypting the content of a message, steganography conceals the very existence of the message by embedding it within another seemingly harmless file such as an image, audio, or video. This makes it an ideal choice for covert communication and digital watermarking. Among the various forms of steganography, image steganography is particularly popular due to the wide availability of image files and the subtle visual changes that can be made without affecting the overall appearance of the image to the human eye.

This project, titled "Image Steganography Using Python", presents a user-friendly desktop application that allows users to hide and extract secret messages within digital images. Built using Python's Tkinter library for the graphical user interface and Pillow (PIL) for image processing, the application provides a seamless and interactive experience for users. The system uses the Least Significant Bit (LSB) steganography technique, which modifies the least significant bits of the pixel values to encode data. This method ensures that the changes in the image are imperceptible to the human eye, thereby maintaining the visual integrity of the carrier image.

With the increasing need for protecting personal, organizational, and governmental data, this project aims to offer a practical and efficient solution for secure communication. Through a clean and intuitive interface, users can easily select an image, enter a secret message, and embed the message into the image. The same interface also allows users to decode the hidden message from a stego image. The application demonstrates the power and flexibility of Python in building real-world security tools and serves as a foundational platform for further exploration into more advanced steganographic and cryptographic techniques.

### STEGANOGRAPHY:

Steganography is the practice of hiding secret information within an ordinary file or object in such a way that the presence of the hidden data is not noticeable. The word steganography comes from the Greek words “steganos” meaning “covered” and “graphia” meaning “writing.” In the digital world, steganography is most commonly used to hide messages inside image, audio, or video files. Unlike cryptography, which scrambles the contents of a message to make it unreadable, steganography conceals the very existence of the message, making it harder for unauthorized users to detect that any secret communication is taking place. One popular method in image steganography is the Least

Significant Bit (LSB) technique, where the smallest bits of image pixel values are replaced with bits of the hidden message. This alteration is so minor that the human eye cannot detect any visible change in the image. Steganography is useful in scenarios where privacy, confidentiality, and undetectable communication are important, such as secure messaging, digital watermarking, and data protection.

## II. EXISTING SYSTEM

Current systems for image steganography typically involve either custom-built software or online platforms that allow users to upload images, embed secret messages, and later extract the hidden content. While these tools achieve the basic goal of steganography, they often come with several limitations. Many of the existing systems lack flexibility and are not user-friendly, especially for non-technical users. A majority of them rely on command-line interfaces, making it difficult for individuals without programming experience to operate them effectively. Additionally, online platforms raise significant security and privacy concerns, as users must upload sensitive images and messages to third-party servers, which can potentially expose their data to unauthorized access. Some applications also have limited format support, accepting only specific image types like PNG or BMP, which restricts user convenience. Moreover, many tools do not provide message encryption before embedding, which means if the stego image is discovered, the message can be easily retrieved and read.

**DISADVANTAGES:**

- No password protection, making it easier for unintended users to decode hidden data.
- Lack of cross-platform support or mobile usability.
- Inefficient image processing that can lead to quality degradation or data corruption, especially in compressed formats like JPEG.

## III. PROPOSED SYSTEM

The proposed system is a desktop-based application for image steganography using Python. It is designed to be user-friendly and accessible even for beginners. The system features a Graphical User Interface (GUI) developed using Tkinter, and image processing is handled by the Pillow (PIL) library. This application allows users to easily embed (encode) and extract (decode) secret messages into/from digital images.

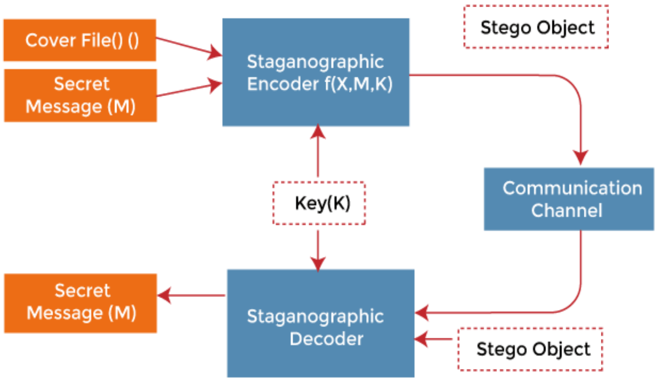
Key features include:

- Simple GUI with buttons to choose image, enter message, encode, and decode.
- Support for common image formats like PNG and BMP.
- Optional encryption/decryption for added data security before embedding.
- Error handling for invalid inputs or unsupported images.
- Lightweight and offline tool – no need for an internet connection.
- Help section or guide to assist new users in using the application.
- Provide clear error handling and help documentation.

**ADVANTAGES:**

- Easy to use: User-friendly GUI with no technical skills required.
- Secure: Optional encryption and local processing ensure privacy.
- Offline functionality: Works without internet; data remains on user’s device.
- Visual integrity: The original look of the image is preserved after encoding.
- Supports decoding even if the user did not encode the image themselves.
- Customizable: Can be extended for future features like password protection or mobile version.

## IV. SYSTEM ARCHITECTURE



## V. METHODOLOGY

1.Requirement Analysis
The goal is to create a user-friendly desktop application that hides and retrieves secret messages in images securely without affecting image quality.

2. Technology Used
Python is used as the programming language. Tkinter is used for building the GUI, and Pillow (PIL) is used for image processing.

3. GUI Design
A simple interface is designed with buttons to select an image, input a message, encode it, decode it, and save the output image. It is designed for users with no technical background.

4. Encoding Process
The message is converted to binary and hidden in the Least Significant Bits (LSB) of the image pixels. The modified image (stego image) is then saved.

5. Decoding Process
The system reads the LSBs from the stego image, converts the bits back to characters, and displays the hidden message.

6. Optional Encryption
For extra security, the message can be encrypted before hiding it, and decrypted after extraction (future enhancement).

7. Testing
The system is tested with various images and messages to ensure correct encoding/decoding, image quality.

**IMPLEMENTATION:**
The Image Steganography system is implemented using Python, and the following libraries and technologies are used:

- Tkinter for the graphical user interface (GUI).
- Pillow (PIL) for image processing (loading, pixel manipulation).
- Cryptography for optional encryption/decryption of messages.
- Python bit manipulation for embedding and extracting messages from image pixels.
- Unit test for unit testing the individual modules.

The implementation consists of:

- A GUI for users to select images and input messages.
- Backend logic for encoding and decoding messages into images.

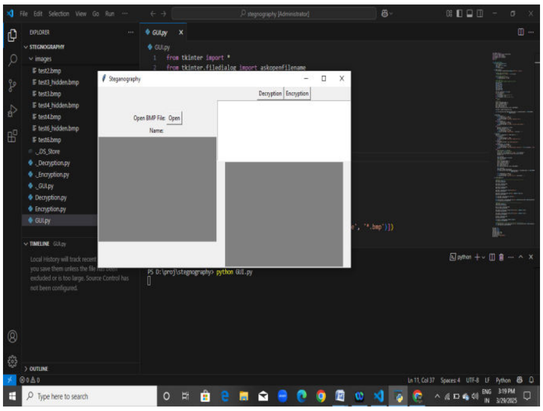
ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- Error handling for invalid inputs and formats.
- Optional encryption for added security.

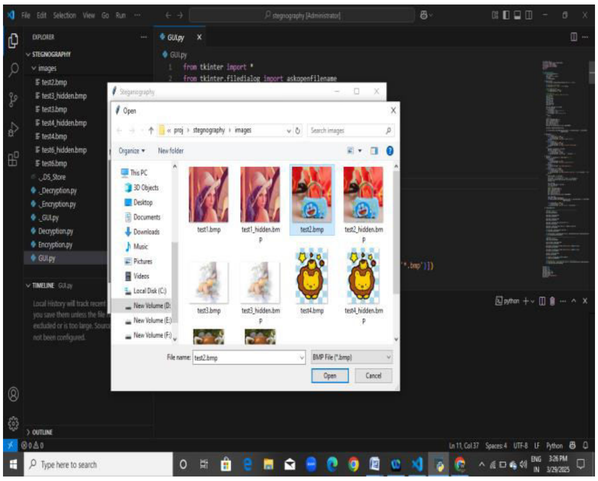
## VI. RESULTS



```
from tkinter import *
from tkinter.filedialog import askopenfilename
```

```
PS D:\proj\stegnography> python GUI.py
```

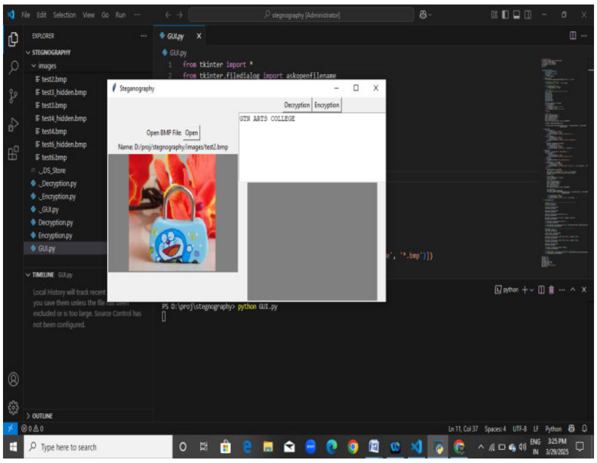
### 1. SELECTING IMAGE



```
from tkinter import *
from tkinter.filedialog import askopenfilename
```

File name: test2.bmp

### 2. ENCRYPTING TEXT MESSAGE



```
from tkinter import *
from tkinter.filedialog import askopenfilename
```

Name: D:/proj/stegnography/images/test2.bmp

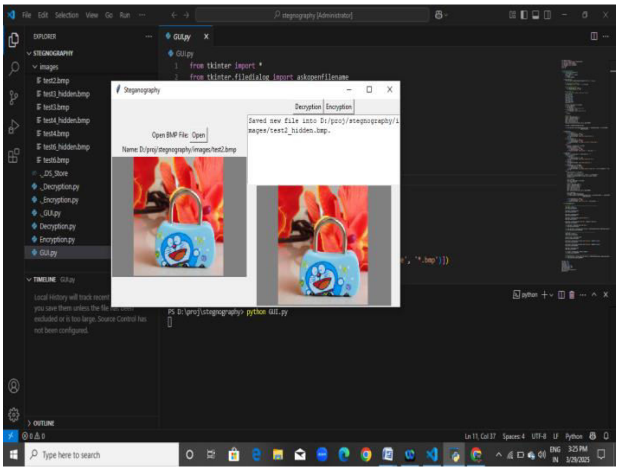
GTN ARTS COLLEGE

```
PS D:\proj\stegnography> python GUI.py
```

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



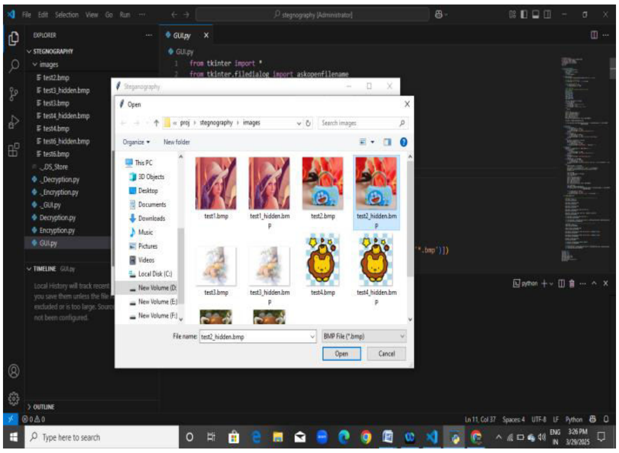
```
from tkinter import *
from tkinter.filedialog import askopenfilename
```

Name: D:/proj/stegnography/images/test2.bmp

```
Saved new file into D:/proj/stegnography/images/test2_hidden.bmp.
```

```
PS D:\proj\stegnography> python GUI.py
```

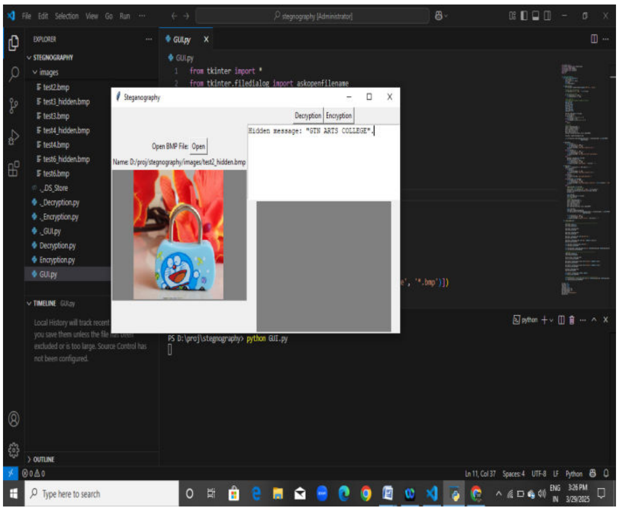
### 3. OPENING ENCRYPTED IMAGE (test2_hidden.bmp)



File name: test2_hidden.bmp

BMP File (*.bmp)

### 4. DECRYPTION OUTPUT



Name: D:/proj/stegnography/images/test2_hidden.bmp

```
Hidden message: "GTN ARTS COLLEGE".
```

```
PS D:\proj\stegnography> python GUI.py
```

## VII. CONCLUSION

The Image Steganography System using Python is a secure and user-friendly tool designed for hiding secret messages within digital images. It enables covert communication by embedding messages without affecting the visual quality of the carrier image, making it ideal for situations where privacy and discretion are essential. Built with Python libraries like Tkinter and Pillow, the system offers a simple graphical interface suitable for all users, including those without technical knowledge. By applying the Least Significant Bit (LSB) technique, it ensures that hidden data remains undetectable to the human eye. The application has been tested under various conditions to ensure reliable performance and accurate message retrieval. With the potential for future enhancements such as encryption, cloud integration, and support for audio and video files, this project provides a strong foundation for developing advanced data security solutions through steganography.

## VIII. FUTURE ENHANCEMENTS

To improve the image steganography system, several future upgrades can be considered. Support for audio and video steganography will allow users to hide messages in multimedia files, increasing security and usability. Cloud integration can enable secure online storage and sharing of stego images, making the tool more accessible in collaborative environments. Adding advanced encryption techniques like AES and RSA will enhance data protection by ensuring that hidden messages remain secure even if intercepted. Finally, cross-platform support through mobile apps and web interfaces will make the system available to a wider audience, allowing secure communication on the go. These features will make the system more robust, versatile, and suitable for real-world use.

## REFERENCES

[1] S. Katzenbeisser and F. A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Norwood, MA, USA, 2000.
[2] N. F. Johnson, Z. Duric, and S. Jajodia, Information Hiding: Steganography and Watermarking—Attacks and Countermeasures, Springer, Boston, MA, USA, 2001. doi: 10.1007/978-1-4615-1287-8.
[3] P. Wayner, Disappearing Cryptography: Information Hiding: Steganography & Watermarking, 3rd ed., Morgan Kaufmann, Burlington, MA, USA, 2009.
[4] A. Nag, K. Dasgupta, and S. Biswas, Steganography: An Art of Hiding Data, Lambert Academic Publishing, Saarbrücken, Germany, 2015.
[5] M. Hussain and M. Hussain, "A Survey of Image Steganography Techniques," International Journal of Advanced Science and Technology, vol. 54, pp. 113–124, 2013.

# INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com